SPoD Application Concept

The following topics are covered:

Conventional Approach

A conventional Natural application consists of a collection of Natural and Non-Natural objects. Together, they form a functional unit which covers the business logic for a particular business problem. An application consists of a set of libraries and their Natural objects. It is possible that one part of the library belongs to one application while another part belongs to another (different) application.

SPoD Approach

With the SPoD approach, the term "Natural Application" is introduced. It provides a **logical view** of Natural and Non-Natural objects (such as job control files). A Natural application does not contain the objects, but it is a **collection of links** to Natural objects or "sub-applications" describing where the objects are stored. Natural objects or "sub-applications" can be linked to several applications.

In a SPoD scenario, the following types of applications exist:

- base application
- compound application

The space where the applications are maintained and displayed at the workstation is called the Application Workspace.

The definitions are stored in the development server file.

Note: Predict Version 4.2 supports this application concept. Earlier versions of Predict cannot be used.

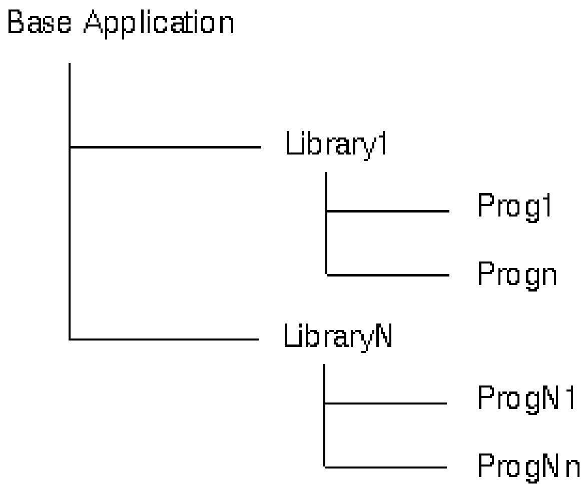
Some Arguments for Using the Term "Natural Application"

The following topics may give you an idea why you should use the term "Natural Application" and the concept that stands behind it:

- It groups the content of your libraries in a logical order based on the different applications you are maintaining that can be displayed in parallel to the library structure without using an additional tool.
- It focusses your view on the number of objects of a library you are intend to work on. You need no longer navigate through all Natural objects located in one library, because you can edit the objects directly from the Application Workspace.
- It reduces the maintenance effort, because you are able to link all objects of different libraries to the application, especially those located in the STEPLIBs.
- It gives you the opportunity to group an application into different "sub-applications" whereas each "sub-application" can be assigned to a member of the development team that is responsible for its development or maintenance.
- Thinking of client/server architecture, you are able to group your application into different "sub-applications", and each "sub-application" can be stored on a different platform at application runtime.

Base Application

A base application is defined as a set of Natural object links. The associated Natural objects pertain to one specific Natural Development Server and are all located on the **same** FUSER system file.



Base Application Structure (Example)

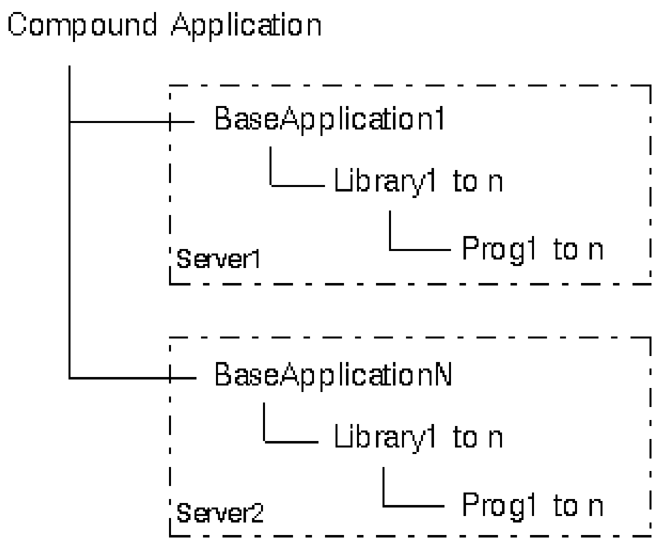
The following information is stored for a base application:

- Name of the application
- Description of the application (textual information only)
- Name and port ID of the Natural development server where the linked objects reside, i.e. the application environment
- Profile for the application environment
- Identification of each object linked

Compound Application

A compound application enables the application developer to combine several base applications. It is defined as a set of links to these base applications.

The base applications involved in a compound application can be spread across **different** FUSER system files or different development servers. Each base application may have a different parameter setting.



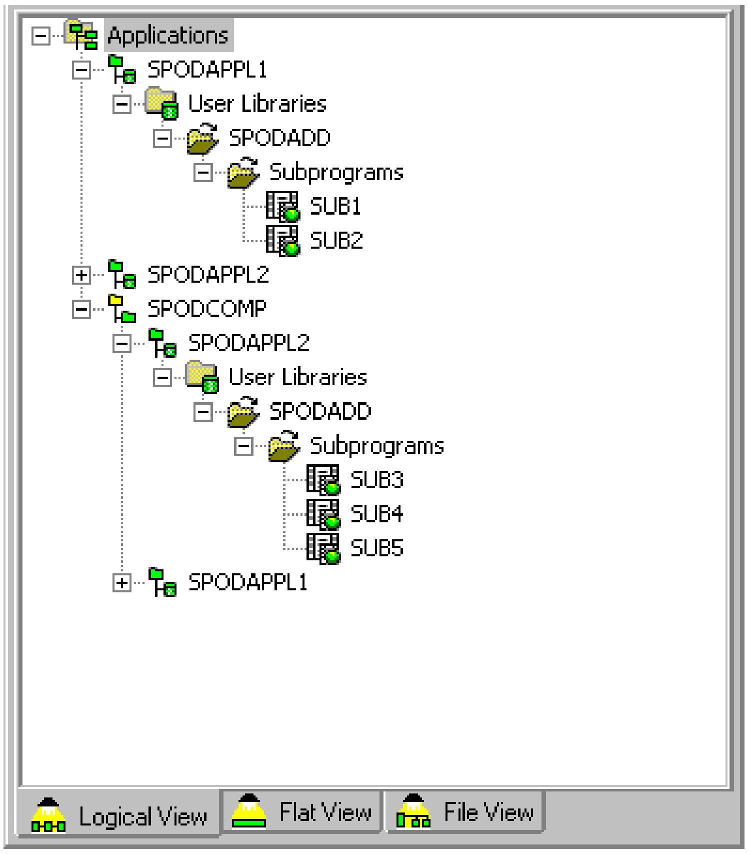
Compound Application Structure (Example)

The following information is stored for a compound application:

- Name of the application
- Description of the application
- Identification of each base application linked

Application Workspace

The Application Workspace is an area on the Natural Studio screen (similar to Natural Studio’s Library Workspace) where all known applications and their objects are shown in a view which complements the existing logical, flat and file views and displays a tree structure comprising all program objects belonging to an application, see example below.



```
Applications
  SPODAPPL1
    User Libraries
      SPODADD
        Subprograms
          SUB1
          SUB2
  SPODAPPL2
  SPODCOMP
    SPODAPPL2
      User Libraries
        SPODADD
          Subprograms
            SUB3
            SUB4
            SUB5
    SPODAPPL1
```